REMARKS

Claims 1-40 were pending in the application. Claims 1-40 were rejected. Claims 7, 9, 18, 27, 29, and 38 are amended. Claims 41-60 are added. Claims 1-60 are now pending in the application. Claims 1, 7, 21, and 27 are the independent claims. Reconsideration of the amended application is respectfully requested.

The Examiner rejected claims 1-6 and 21-26 under 35 USC §103(a) as being unpatentable over Lipner et al.

Independent claim 1 recites a method of encrypting an object. According to the recited method, a plurality of key splits is combined to generate a cryptographic key. A cryptographic algorithm is initialized with the cryptographic key. The initialized cryptographic algorithm is applied to the object, to form an encrypted object. At least one of the plurality of key splits corresponds at least in part to a biometric measurement.

The Examiner cited Lipner et al. as disclosing that a plurality of key splits is combined to generate a cryptographic key, citing column 15, lines 13-16. This passage discusses a multiple split session key, referring to Fig. 13, which shows the flow of a sending program. According to this program, a secret session key KS is negotiated and formed, and then used to encrypt a message M. It is not disclosed that this key is formed through the combination of key splits. Only after encryption of the message, the session key is split into multiple parts. The example mentions splitting the key into two halves KS1 and KS2. See column 14, lines 55-66. These are the split session keys mentioned in the passage cited by the Examiner.

The Examiner also cited Lipner et al. column 7, lines 40-43 as disclosing initialization of a cryptographic algorithm with a cryptographic key. The passage cited by the Examiner states only that the Lipner et al. invention uses an unclassified data encryption algorithm. The passage does not describe initializing the algorithm with a cryptographic key, and particularly not with a cryptographic key formed by combining splits.

In rebuttal, the Examiner further stated that Lipner et al. disclose that a private key is generated from combined components in the passage at column 13, lines 43-50. This passage describes the function of the law enforcement decryptor component of the Lipner et al. invention. According to this passage, private key components are combined to form a private key used to decrypt a session key. The decrypted session key is used to decrypt the message. See column 13, lines 48-54. This gives law enforcement agents the ability to decrypt an encrypted message using only components found in a law enforcement header. Claim 1 recites a process that includes encrypting an object using an algorithm initialized by a key formed from key splits, not a decryption process enabled by a combined key.

The Examiner stated that the "basis, or compliment of this private key is of course used to first encrypt an object in accordance with the well established public key/private key algorithm." It is respectfully submitted that this is not the case. At column 10, lines 52-57, it is disclosed that the private key halves are split from the private key after the private key is formed, as part of the process of generating program parameters. Thus, the private key is not generated from key splits; rather, the private key halves are generated

from the private key. Further, as described at column 12, lines 13-20, the message is encrypted using a secret session key. The session key itself is encrypted using a public key, which is why it is later decrypted using the combined private key, as pointed out by the examiner. This is typical of public key cryptography; an object decrypted using a private key, split or otherwise, was encrypted using a public key. Use of a combined private key to decrypt an object does not imply that the combined private key was used to encrypt the object, and in fact the opposite is implied.

The Examiner asserted that splitting a key and combining a key are commutative functions, and that if one can split a key, one could also form a key from splits. This might or might not be true, depending on the function used to create the key from the splits, but in any case Lipner et al. do not disclose generating a key from splits for use in encrypting an object, as required by claim 1. That is, even if Lipner et al. have the ability to form a key from splits, that ability is not exercised by Lipner et al., who also do not suggest any reason or motivation for deriving the keys from splits prior to encryption. Lipner et al. split a private key for the particular purpose of allowing a law enforcement agency to gain access to an encrypted message, that is, for purposes of decryption. Lipner et al. do not show any purpose for generating the corresponding public key from key splits for purposes of encryption.

The Examiner stated that Lipner et al. disclose generating the key halves first, at column 14, lines 30-35. This passage describes a different embodiment than that described above. The description of this second embodiment begins at column 13, line 55. In this embodiment, two private keys and two public keys are used (column 14, lines

20-24), as keys from two different respective escrow agents (column 14, lines 41-45). These keys are obtained from an external source, or are generated by a key escrow entity (column 14, lines 28-40). A message is encrypted using a secret session key, which is then split into halves (column 14, lines 55-66). Each half of the session key is encrypted using the public key of one of the two escrow agents to form the law enforcement header (column 14, line 67-column 15, line3). Thus, what the Examiner refers to as key halves are actually separate keys belonging to different escrow agents. These keys are included in the law enforcement header, used by law enforcement agents to decrypt the session key so that the encrypted message can be accessed (column 15, lines 58-62). The session key is split into halves, but only after encrypting the message, as noted above. The Examiner stated that he interprets the teachings of Lipner et al. to suggest that a key can be formed and split and that two keys can be created and joined in some fashion. However, Lipner et al. actually teach that a session key can be split after encrypting a message, and that separate keys from separate agents can be used to encrypt and decrypt the session key halves, which can be combined to decrypt the message. In contrast, claim 1 recites a process in which key splits are combined to generate a key, which is used to initialize an algorithm that is applied to an object to encrypt the object. Lipner et al. do no disclose or suggest this process, and teach no reason for modifying their process in order to provide these actions.

Lipner et al. disclose use of biometric tests for authentication, as noted by the Examiner, at column 21, lines 34-41. These tests are used as a replacement for, or in addition to, password-type authentication for access to a system. Lipner et al. do not

disclose or suggest use of biometric correspondence to a cryptographic key split. Lipner et al. disclose that a public/private key pair KU can be seeded with external parameters, but there is no suggestion that biometric authentication data, as disclosed by Lipner et al., would correspond to an actual key split used to form the key, particularly since Lipner et al. does not form cryptographic keys from key splits.

In rebuttal, the Examiner acknowledged that Lipner et al. do not explicitly disclose using biometric information as one part of the key, but asserted that the passage at column 14, lines 30-35 teaches that keys are generated and initialized in parts. However, that passage only describes that separate public keys and separate private keys are generated, corresponding to respective separate escrow agents. These keys are never combined; rather, they are used separately to decrypt halves of the session key, which had previously been used to encrypt the message (column 14, lines 55-61; column 15, line 3; column 15, lines 58-62). Thus, the public and private keys are not generated in parts; they are separate keys that never get combined.

The Examiner also asserted that Lipner et al. explicitly teaches, at column 10, lines 50-52, that keys can be seeded with externally-generated parameters, and asserted that one skilled in the art knows that such a seed must be random. The Examiner further stated that a person's biometric data would be "totally different and random from another person," and asserted that one would be motivated to use such a personal way of creating a totally random key because it is very unlikely that another person can duplicate someone else's biometric data. The Examiner concluded that one of ordinary skill in the art would have been motivated to modify the teachings of Lipner et al. to use a biometric

device to generate a portion of the private key because it is an external random source of input.

Initially, it is respectfully pointed out that a biometric input is not a random input. Claim 1 recites that at least one of a plurality of key splits corresponds at least in part to a biometric measurement. A random input is one that has no specific pattern, purpose, or objective; it is unsystematically generated, and all outcomes are equally likely. A biometric measurement is not random; it is based on a biological characteristic of a person. A biometric measurement can be repeated, and the results of repeated measurements are substantially identical, within a reasonable tolerance. If a system requires a random input, a biometric measurement would not suffice.

Further, the passage cited by the Examiner discusses seeding a program-unique key with a parameter. This key is later split. Lipner et al. do not disclose forming the key from splits, one of which includes correspondence with a biometric measurement. Rather, Lipner et al. disclose generating the key, seeding the key with a parameter, and then splitting the key for future use (column 10, lines 47-57). The only use that Lipner et al. disclose for a biometric measurement is for authentication purposes (column 21, lines 34-42); Lipner et al. do not disclose use of a biometric measurement for correspondence to a key, and Lipner et al. do not disclose or suggest any benefit for doing so.

For at least the foregoing reasons, the invention as recited in claim 1 is not rendered obvious by Lipner et al. Claims 2-6 depend from claim 1, and therefore also are not rendered obvious by Lipner et al., for the reasons stated above, and also because of the additional features these claims recite.

For example, claim 2 recites adding at least one key split to the encrypted object. The Examiner stated that Lipner et al., at column 15, lines 31-32, add a key split (ELVS) to the encrypted message. In the paragraph previous to the passage cited by the Examiner, ELVS is identified as an encrypted leaf verification string, not a key split. LVS, the unencrypted form of ELVS, includes both halves of the session key. However, as pointed out previously, the session key is split only after encrypting the message (column 14, lines 55-63). Thus, ELVS cannot be fairly characterized as a key split, and certainly not a key split that was used to form a key that was used to encrypt an object, as required by claim 2, because ELVS is formed only after formation of the key itself.

Claim 3 recites adding reference data associated with at least one key split to the encrypted object. The Examiner stated that Lipner et al., at column 15, lines 31-32, add reference data (LEAF) to the encrypted message. LEAF includes both halves of the session key. However, as pointed out previously, the session key is split only after encrypting the message (column 14, lines 55-63). Thus, LEAF cannot be fairly characterized as being associated with a key split that was used to form a key that was used to encrypt an object, as required by claim 3.

Claim 4 recites retrieving at least one of the plurality of key splits from a storage medium. The Examiner cited Lipner et al., at column 10, lines 35-38, as teaching retrieval of key splits from memory. The cited passage discloses obtaining the private component of a public/private key pair from memory. A private key is a fully-formed key; it is one key of a pair of keys used for secure communication. It is not a key split, and the particular key noted by the Examiner in this passage is not the same entity that

the Examiner has been asserting as the key split in other portions of the rejection. The private key cited by the Examiner is not combined with a key split to generate a key, as required by claim 4.

Claim 5 recites that the storage medium of claim 4 is disposed on a smart card. The Examiner noted that Lipner et al. disclose that encryption can be performed on hardware such as a PCMCIA card. The Examiner asserted that it is inherent that the smartcard has storage capabilities necessary for performing encryption, and that combining the key splits is performed on a smart card. Claim 5 does not recite that the smartcard has storage capabilities necessary for performing encryption, or that key splits are combined on the smart card. Claim 5 recites that a storage medium from which at least one key split is retrieved is disposed on a smart card. Lipner et al. disclose that encryption can be performed on hardware such as a PCMCIA card, but do not disclose that a key split used to form an encryption key prior to encryption is retrieved from a storage medium disposed on a smart card, as required by claim 5.

Claim 6 recites that combining a plurality of key splits to generate a cryptographic key is performed on a smart card. The Examiner noted that Lipner et al. disclose that encryption can be performed on hardware such as a PCMCIA card. The Examiner asserted that it is inherent that combining the key splits is performed on a smart card. It is respectfully submitted that this assertion is not inherent. A smart card can include the processing capability for performing encryption, including the generation of control signals for performing certain functions at another processor in communication with the smart card. No inherent processing arrangement is implied by including a smart card in

the system, and Lipner et al. do not teach or suggest the combining of key splits on a smart card, as required by claim 6.

For at least the reasons noted above, the rejection of claims 1-6 should be withdrawn.

Independent claim 21 recites a storage medium that includes instructions for causing a data processor to encrypt an object. The instructions include generate a cryptographic key by combining a plurality of key splits, initialize a cryptographic algorithm with the cryptographic key, and apply the initialized cryptographic algorithm to the object to form an encrypted object. At least one of the key splits corresponds at least in part to a biometric measurement.

The Examiner cited Lipner et al. as disclosing that a plurality of key splits is combined to generate a cryptographic key, citing column 15, lines 13-16. This passage discusses a multiple split session key, referring to Fig. 13, which shows the flow of a sending program. According to this program, a secret session key KS is negotiated and formed, and then used to encrypt a message M. It is not disclosed that this key is formed through the combination of key splits. After encryption of the message, the session key is split into multiple parts. The example mentions splitting the key into two halves KS1 and KS2. See column 14, lines 55-66. These are the split session keys mentioned in the passage cited by the Examiner.

The Examiner also cited Lipner et al. column 7, lines 40-43 as disclosing initialization of a cryptographic algorithm with a cryptographic key. The passage cited by the Examiner states that the Lipner et al. invention uses an unclassified data

encryption algorithm. The passage does not describe initializing the algorithm with a cryptographic key, and particularly not with a cryptographic key formed by combining splits.

In rebuttal, the Examiner further stated that Lipner et al. disclose that a private key is generated from combined components in the passage at column 13, lines 43-50. This passage describes the function of the law enforcement decryptor component of the Lipner et al. invention. According to this passage, private key components are combined to form a private key used to decrypt a session key. The decrypted session key is used to decrypt the message. See column 13, lines 48-54. This gives law enforcement agents the ability to decrypt an encrypted message using only components found in a law enforcement header. Claim 21 recites instructions for causing a processor to encrypt an object according to an algorithm initialized by a key formed from key splits, not a decryption process enabled by a combined key.

The Examiner stated that the "basis, or compliment of this private key is of course used to first encrypt an object in accordance with the well established public key/private key algorithm." It is respectfully submitted that this is not the case. At column 10, lines 52-57, it is disclosed that the private key halves are split from the private key after the private key is formed, as part of the process of generating program parameters. Thus, the private key is not generated from key splits; rather, the private key halves are generated from the private key. Further, as described at column 12, lines 13-20, the message is encrypted using a secret session key. The session key itself is encrypted using a public key, which is why it is later decrypted using the combined private key, as pointed out by

the examiner. This is typical of public key cryptography; an object decrypted using a private key, split or otherwise, was encrypted using a public key. Use of a combined private key to decrypt an object does not imply that the combined private key was used to encrypt the object, and in fact the opposite is implied, that is, that a public key was used to encrypt the object.

The Examiner asserted that splitting a key and combining a key are commutative functions, and that if one can split a key, one could also form a key from splits. This might or might not be true, but Lipner et al. does not in fact disclose generating a key from splits for use in encrypting an object. Lipner et al. split a private key for the particular purpose of allowing a law enforcement agency to gain access to an encrypted message. Lipner et al. do not disclose, or show any purpose for, generating the corresponding public key from key splits before performing the encryption.

The Examiner stated that Lipner et al. disclose generating the key halves first, at column 14, lines 30-35. This passage describes a different embodiment than that described above. The description of this second embodiment begins at column 13, line 55. In this embodiment, two private keys and two public keys are used (column 14, lines 20-24), as keys from two different escrow agents (column 14, lines 41-45). These keys are obtained from an external source, or are generated by a key escrow entity (column 14, lines 28-40). A message is encrypted using a secret session key, which is then split into halves (column 14, lines 55-66). Each half of the session key is encrypted using the public key of one of the two escrow agents to form the law enforcement header (column 14, line 67-column 15, line3). Thus, what the Examiner refers to as key halves are

actually separate keys belonging to different escrow agents. These keys are included in the law enforcement header, used by law enforcement agents to decrypt the session key so that the encrypted message can be accessed (column 15, lines 58-62). The session key is split into halves, but only after encrypting the message, as noted above. The Examiner stated that he interprets the teachings of Lipner et al. to suggest that a key can be formed and split and that two keys can be created and joined in some fashion. However, Lipner et al. actually teach that a session key can be split after encrypting a message, and that separate keys from separate agents can be used to encrypt and decrypt the session key halves, which can be combined to decrypt the message. In contrast, claim 1 recites a process in which key splits are combined to generate a key, which is used to initialize an algorithm that is applied to an object to encrypt the object. Lipner et al. do no disclose or suggest this process, and teach no reason for modifying their process in order to provide these actions.

Lipner et al. disclose use of biometric tests for authentication, as noted by the Examiner, at column 21, lines 34-41. These tests are used as a replacement for, or in addition to, password-type authentication for access to a system. Lipner et al. do not disclose or suggest use of biometric correspondence to a cryptographic key split. Lipner at al. disclose that a public/private key pair KU can be seeded with external parameters, but there is no suggestion that biometric authentication data, as disclosed by Lipner et al., would correspond to an actual key split used to form the key, particularly since Lipner et al. does not form cryptographic keys from key splits.

In rebuttal, the Examiner acknowledged that Lipner et al. do not explicitly disclose using biometric information as one part of the key, but asserted that the passage at column 14, lines 30-35 teach that keys are generated and initialized in parts. However, that passage only describes that separate public keys and separate private keys are generated, corresponding to respective separate escrow agents. These keys are never combined; rather, they are used separately to decrypt halves of the session key, which had previously been used to encrypt the message (column 14, lines 55-61; column 15, line 3; column 15, lines 58-62). Thus, the public and private keys are not generated in parts; they are separate keys that never get combined.

The Examiner also asserted that Lipner et al. explicitly teaches, at column 10, lines 50-52, that keys can be seeded with externally-generated parameters, and asserted that one skilled in the art knows that such a seed must be random. The Examiner further stated that a person's biometric data would be "totally different and random from another person," and asserted that one would be motivated to use such a personal way of creating a totally random key because it is very unlikely that another person can duplicate someone else's biometric data. The Examiner concluded that one of ordinary skill in the art would have been motivated to modify the teachings of Lipner et al. to use a biometric device to generate a portion of the private key because it is an external random source of input.

Initially, it is respectfully pointed out that a biometric input is not a random input. Claim 21 recites that at least one of a plurality of key splits corresponds at least in part to a biometric measurement. A random input is one that has no specific pattern, purpose, or

objective; it is unsystematically generated, and all outcomes are equally likely. A biometric measurement is not random; it is based on a biological characteristic of a person. A biometric measurement can be repeated, and the results of repeated measurements are substantially identical, within a reasonable tolerance. If a system requires a random input, a biometric measurement would not suffice.

Further, the passage cited by the Examiner discusses seeding a program-unique key with a parameter. This key is later split. Lipner et al. do not disclose forming the key from splits, one of which includes correspondence with a biometric measurement. Rather, Lipner et al. disclose generating the key, seeding the key with a parameter, and then splitting the key for future use (column 10, lines 47-57). The only use that Lipner et al. disclose for a biometric measurement is for authentication purposes (column 21, lines 34-42); Lipner et al. do not disclose use of a biometric measurement for correspondence to a key, and Lipner et al. do not disclose or suggest any benefit for doing so.

For at least the foregoing reasons, the invention as recited in claim 21 is not rendered obvious by Lipner et al. Claims 22-26 depend from claim 21, and therefore also are not rendered obvious by Lipner et al., for the reasons stated above, and also because of the additional features these claims recite.

For example, claim 22 recites instructions to add at least one key split to the encrypted object. The Examiner stated that Lipner et al., at column 15, lines 31-32, add a key split (ELVS) to the encrypted message. In the paragraph previous to the passage cited by the Examiner, ELVS is identified as an encrypted leaf verification string, not a key split. LVS, the unencrypted form of ELVS, includes both halves of the session key.

However, as pointed out previously, the session key is split only after encrypting the message (column 14, lines 55-63). Thus, ELVS cannot be fairly characterized as a key split, and certainly not a key split that was used to form a key that was used to encrypt an object, as required by claim 22.

Claim 23 recites instructions to add reference data associated with at least one key split to the encrypted object. The Examiner stated that Lipner et al., at column 15, lines 31-32, add reference data (LEAF) to the encrypted message. LEAF includes both halves of the session key. However, as pointed out previously, the session key is split only after encrypting the message (column 14, lines 55-63). Thus, LEAF cannot be fairly characterized as being associated with a key split that was used to form a key that was used to encrypt an object, as required by claim 23.

Claim 24 recites instructions to retrieve at least one of the plurality of key splits from a memory. The Examiner cited Lipner et al., at column 10, lines 35-38, as teaching retrieval of key splits from memory. The cited passage discloses obtaining the private component of a public/private key pair from memory. A private key is a fully-formed key; it is one key of a pair of keys used for secure communication. It is not a key split, and the particular key noted by the Examiner in this passage is not the same entity that the Examiner has been asserting as the key split in other portions of the rejection. The private key cited by the Examiner is not combined with a key split to generate a key, as required by claim 24.

Claim 25 recites that the memory of claim 24 is disposed on a smart card. The Examiner noted that Lipner et al. disclose that encryption can be performed on hardware

such as a PCMCIA card. The Examiner asserted that it is inherent that the smartcard has storage capabilities necessary for performing encryption, and that combining the key splits is performed on a smart card. Claim 25 does not recite that the smartcard has storage capabilities necessary for performing encryption, or that key splits are combined on the smart card. Claim 25 recites that a storage medium from which at least one key split is retrieved is disposed on a smart card. Lipner et al. disclose that encryption can be performed on hardware such as a PCMCIA card, but do not disclose that a key split used to form an encryption key prior to encryption is retrieved from a memory disposed on a smart card, as required by claim 25.

Claim 26 recites that the data processor is distributed, and the instruction to generate a cryptographic key is executed at least in part on a smart card. The Examiner noted that Lipner et al. disclose that encryption can be performed on hardware such as a PCMCIA card. The Examiner asserted that it is inherent that combining the key splits is performed on a smart card. It is respectfully submitted that this assertion is not inherent, or necessarily true. A smart card can include the processing capability for performing encryption, including the generation of control signals for performing certain functions at another processor in communication with the smart card. No inherent processing arrangement is implied by including a smart card in the system, and Lipner et al. do not teach or suggest the distributed nature of the processor or the combining of key splits on a smart card, as required by claim 26.

For at least the reasons noted above, the rejection of claims 21-26 should be withdrawn.

The Examiner rejected claims 7, 8, 10-28 and 30-40 under 35 USC §103(a) as being unpatentable over Sudia, in view of Lipner et al.

Independent claim 7 recites a method of encrypting an object by a user, in a cryptographic system associated with an organization. According to the claimed method, a first cryptographic key is generated by combining an organization split corresponding to the organization, a maintenance split, a random split, and at least one label split. A cryptographic algorithm is initialized with the first cryptographic key. The object is encrypted according to the initialized cryptographic algorithm. Combiner data is added to the encrypted object. The combiner data includes reference data corresponding to at least one of the at least one label split and the cryptographic algorithm, name data associated with the organization, the maintenance split and/or a maintenance level associated with the maintenance split, and the random split. The encrypted object is stored with the added combiner data

The Examiner cited Sudia at column 18, lines 65-67 as disclosing generation of a private key by combining key splits. That passage discusses the possibility of recombining splits of a private key in order to reassemble the private key. However, as noted, the private key is reassembled after the splits are recombined. The key is initially generated by a trusted device (column 17, line 65 - column 18, line 3). Independent claim 7 recites generating a cryptographic key by combining key splits. As described in the section leading into the passage cited by the Examiner (column 17, line 29 through column 18, line 61), the disclosed invention utilizes a public/private encryption key pair. The private key is merely generated randomly by a firmware program (column 17, lines

50-61). Later, the generated private key is divided into a number of splits (column 18, lines 12-15) after the message has been encrypted and transmitted (column 17, line 65 through column 18, line 11).

Further, claim 7 specifies that the key splits used to generate the key include an organization split corresponding to the organization, a maintenance split, a random split, and at least one label split. Sudia does not disclose these particular splits.

The Examiner also cited column 10, lines 62-67 as disclosing the initialization of a cryptographic algorithm using a key. However, that passage merely describes that a known method is used to split private encryption keys into components. Sudia does not disclose using a key generated from the specific splits recited in claim 7 to initialize a cryptographic algorithm.

The Examiner cited column 11, lines 33-39 as disclosing the addition of combiner data. That passage describes the use of a message control header that includes particular law enforcement information, but is not disclosed to include the reference data corresponding to key splits recited in claim 7. Likewise, Fig. 18 shows the message control header as described, including exemplary header components, but makes no correspondence between this data and key splits that are used to generate a key, as required by claim 7.

The Examiner stated that Sudia teaches that the private key is broken up into splits. Claim 7 does not recite this feature. Claim 7 recites that specific splits are combined to generate a cryptographic key.

The Examiner stated that Sudia teaches, at column 18, lines 12-61, that a random number is generated for each split and, consequently, a random number is associated with each key. This passage describes that after the key is split, a random number is generated for each key split of the private key. The key split, the random number, and other information are then assembled into a share packet. This is not the same as providing a random split as a component of a generated key, as required by claim 7. By the time the random number is associated with the key split in the Sudia system, the key has already been generated by the trusted device (column 17, line 65-column 18, line 3). The key split that is associated with the random number comes from a key that has already been generated; the random number can't be a split used to generate the key, because the key has already been generated. The key splits are later reassembled by law enforcement to reform the private key so that an encrypted communication can be accessed. The random numbers are used to assist in verifying that the splits are valid and that the reassembled key is authentic (column 29, line 34-column 30, line 19). The random number is not itself a split used to generate the key, as required by claim 7.

The Examiner acknowledged that Sudia does not disclose an organizational split, a maintenance split, or a label split. The Examiner stated that Lipner et al. discloses seeding a key with externally-generated parameters, and that Sudia discloses organization, maintenance, and label data. The Examiner asserted that these teachings in combination are sufficient to teach one of ordinary skill in the art to provide the data as key splits when generating the key. However, neither reference discloses generating a key from splits at all. Each reference discloses generating a key, and then later splitting

the key, after encrypting a message. Claim 7 recites generating a key from splits, then encrypting an object. Neither reference discloses or suggests any benefit to generating a key from these specific splits; both references only disclose benefits of a key management scheme in which keys are split after encrypting a message. In any case, the Examiner previously asserted that one of ordinary skill in the art would know that the seed data described by Lipner et al. must be a random number, which is inconsistent with the Examiner's assertion here.

Thus, the Sudia invention is fundamentally different from the claimed invention. Some differences between Lipner et al. and the claimed invention are noted above. Neither reference satisfies the deficiencies of the other reference, with respect to the claimed invention. For at least all of the stated reasons, no combination of the teachings of Sudia and Lipner et al. could render obvious the invention recited in claim 7. Claims 8 and 10-20 depend from claim 7 and therefore also are not rendered obvious by the asserted combination, for the reasons noted above, and also because of the additional features recited in the dependent claims.

For example, claim 8 recites selecting the at least one label split from at least one credential, and claim 10 recites that at least one credential is retrieved from a memory. The Examiner stated that Sudia teaches, at column 16, line 31-column 17, line 26, that various types of credentials are stored in memory for each user. It is respectfully pointed out that in this passage, Sudia discloses parameters related to the hardware chip used by Sudia to perform the security function, which are not user credentials (column 16, lines 31-34 and lines 53-56). The Examiner asserted that it is inherent that parameters are

selected from these credentials to seed key splits. Previously, the Examiner asserted that one of ordinary skill in the art would know that the seed data disclosed by Lipner et al. must be random data. User data is not random data, and so the Examiner's discussion is inconsistent in this regard. Further, claims 8 and 10 do not recite seeding key splits. Rather, these claims recite that credentials are selected as label splits, used to generate the key. It is respectfully submitted that disclosures of seeding keys and storing device parameters does not inherently teach one of ordinary skill in the art that a label key split can be selected from a credential.

Claim 11 recites that the memory is disposed on a smart card. The Examiner stated that Sudia teaches, at column 22, lines 63-66, that smart cards contain valuable user identification, and that it would have been obvious to store user information in the memory of a smart card. The passage cited by the Examiner does not mention smart cards, or any other hardware processing device or token. It discusses a system certificate issued to a user by a certifying authority, and reasons for revoking a certificate. Sudia does not disclose memory for storing label splits on a smart card.

Claim 12 recites generating a time stamp corresponding to a time at which the object was encrypted, wherein the combiner data further includes the time stamp. The Examiner cited column 11, lines 33-39 and Fig. 18 of Sudia as disclosing adding combiner data in the form of a message control header that includes a timestamp. Fig. 18 shows that the message control header includes an optional timestamp, but does not disclose that this timestamp is related in any way to the time that an object was encrypted.

Claim 13 recites that the combiner data further includes a user ID associated with the user. The Examiner cited column 11, lines 33-39 and Fig. 18 of Sudia as disclosing adding combiner data in the form of a message control header that includes a an escrow certificate, which in turn includes a user ID in the form of a user name. According to Fig. 18, the message control header does not include a user escrow certificate. Rather, the header includes only the escrow certificate number. It would not be efficient to transmit the entire escrow certificate with an encrypted message, and Sudia does not disclose or suggest doing so.

The Examiner did not specifically address claims 14-17.

Claim 18 recites generating a second cryptographic key based at least in part on the at least one label split, and encrypting the random split with the second cryptographic key, prior to adding the combiner data to the encrypted object. The random split included in the combiner data is the encrypted random split. The Examiner asserted numerous teachings of Sudia, none of which includes a disclosure that a second key is generated that is based on a label split used to generate the first key, or that the second key is used to encrypt the random split used to generate the first key, both prior to adding the combiner data to the encrypted object, and all as required by claim 18.

Claim 19 recites encrypting at least a portion of the combiner data with a header split before adding the combiner data to the encrypted object. Claim 20 recites that the header split is constant. The Examiner stated that Sudia teaches, at column 23, lines 27-31, that part of the header data is encrypted. In that passage, Sudia discloses that the

sender's escrow certificate number is encrypted using the public encryption key. Sudia does not disclose use of a header split for encrypting combiner data.

For at least the reasons noted above, the rejection of claims 7, 8, and 10-20 should be withdrawn.

Independent claim 27 recites a storage medium comprising instructions for causing a data processor to encrypt an object. The instructions include generate a first cryptographic key by combining an organization split corresponding to an organization, a maintenance split, a random split, and at least one label split; initialize a cryptographic algorithm with the first cryptographic key; apply the initialized cryptographic algorithm to the object to form an encrypted object; add combiner data to the encrypted object and store the encrypted object with the combiner data for subsequent access. The combiner data includes reference data corresponding to at least one of the at least one label split and the cryptographic algorithm, name data associated with the organization, at least one of the maintenance split and a maintenance level corresponding to the maintenance split, and the random split.

The Examiner cited Sudia at column 18, lines 65-67 as disclosing generation of a private key by combining key splits. That passage discusses the possibility of recombining splits of a private key in order to reassemble the private key. However, as noted, the private key is reassembled after the splits are recombined. Independent claim 27 recites instructions to generate a cryptographic key by combining key splits. As described in the section leading into the passage cited by the Examiner (column 17, line 29 through column 18, line 61), the disclosed invention utilizes a public/private

encryption key pair. The private key is merely generated randomly by a firmware program (column 17, lines 50-61). Later, the generated private key is divided into a number of splits (column 18, lines 12-15) after the message has been encrypted and transmitted (column 17, line 65 through column 18, line 11).

Further, claim 27 specifies that the key splits used to generate the key include an organization split corresponding to the organization, a maintenance split, a random split, and at least one label split. Sudia does not disclose these particular splits.

The Examiner also cited column 10, lines 62-67 as disclosing the initialization of a cryptographic algorithm using a key. However, that passage merely describes that a known method is used to split private encryption keys into components. Sudia does not disclose using a key generated from the specific splits recited in claim 27 to initialize a cryptographic algorithm.

The Examiner cited column 11, lines 33-39 as disclosing the addition of combiner data. That passage describes the use of a message control header that includes particular law enforcement information, but is not disclosed to include the reference data corresponding to key splits recited in claim 27. Likewise, Fig. 18 shows the message control header as described, including exemplary header components, but makes no correspondence between this data and key splits that are used to generate a key, as required by claim 27.

The Examiner stated that Sudia teaches that the private key is broken up into splits. Claim 27 does not recite this feature. Claim 27 recites instructions for specific splits to be combined to generate a cryptographic key.

The Examiner stated that Sudia teaches, at column 18, lines 12-61, that a random number is generated for each split and, consequently, a random number is associated with each key. This passage describes that after the key is split, a random number is generated for each key split of the private key. The key split, the random number, and other information are then assembled into a share packet. This is not the same as providing a random split as a component of a generated key, as required by claim 27. By the time the random number is associated with the key split, the key has already been generated by the trusted device (column 17, line 65-column 18, line 3). The key split that is associated with the random number comes from a key that has already been generated; the random number can't be a split used to generate the key, because the key has already been generated. The key splits are later reassembled by law enforcement to reform the private key so that an encrypted communication can be accessed. The random numbers are used to assist in verifying that the splits are valid and that the reassembled key is authentic (column 29, line 34-column 30, line 19). The random number is not itself a split used to generate the key, as required by claim 27.

The Examiner acknowledged that Sudia does not disclose an organizational split, a maintenance split, or a label split. The Examiner stated that Lipner et al. discloses seeding a key with externally-generated parameters, and that Sudia discloses organization, maintenance, and label data. The Examiner asserted that these teachings in combination are sufficient to teach one of ordinary skill in the art to provide the data as key splits when generating the key. However, neither reference discloses generating a key from splits at all. Each reference discloses generating a key, and then later splitting

the key, after encrypting a message. Claim 27 recites instructions to generate a key from splits, then encrypt an object. Neither reference discloses or suggests any benefit to generating a key from these specific splits; both references only disclose benefits of a key management scheme in which keys are split after encrypting a message. In any case, the Examiner previously asserted that one of ordinary skill in the art would know that the seed data described by Lipner et al. must be a random number, which is inconsistent with the Examiner's assertion here.

Thus, the Sudia invention is fundamentally different from the claimed invention. Some differences between Lipner et al. and the claimed invention are noted above. Neither reference satisfies the deficiencies of the other reference, with respect to the claimed invention. For at least all of the stated reasons, no combination of the teachings of Sudia and Lipner et al. could render obvious the invention recited in claim 27. Claims 28 and 30-40 depend from claim 27 and therefore also are not rendered obvious by the asserted combination, for the reasons noted above, and also because of the additional features recited in the dependent claims.

For example, claim 28 recites instructions to select the at least one label split from at least one credential, and claim 30 recites instructions to retrieve at least one credential from a memory. The Examiner stated that Sudia teaches, at column 16, line 31-column 17, line 26, that various types of credentials are stored in memory for each user. It is respectfully pointed out that in this passage, Sudia discloses parameters related to the hardware chip used by Sudia to perform the security function, which are not user credentials (column 16, lines 31-34 and lines 53-56). The Examiner asserted that it is

inherent that parameters are selected from these credentials to seed key splits. Previously, the Examiner asserted that one of ordinary skill in the art would know that the seed data disclosed by Lipner et al. must be random data. User data is not random data, and so the Examiner's discussion is inconsistent in this regard. Further, claims 28 and 30 do not recite seeding key splits. Rather, these claims recite that credentials are selected as label splits, used to generate the key. It is respectfully submitted that disclosures of seeding keys and storing device parameters does not inherently teach one of ordinary skill in the art that a label key split can be selected from a credential.

Claim 31 recites that the memory is disposed on a smart card. The Examiner stated that Sudia teaches, at column 22, lines 63-66, that smart cards contain valuable user identification, and that it would have been obvious to store user information in the memory of a smart card. The passage cited by the Examiner does not mention smart cards, or any other hardware processing device or token. It discusses a system certificate issued to a user by a certifying authority, and reasons for revoking a certificate. Sudia does not disclose memory for storing label splits on a smart card.

Claim 32 recites instructions to generate a time stamp corresponding to a time at which the object was encrypted, wherein the combiner data further includes the time stamp. The Examiner cited column 11, lines 33-39 and Fig. 18 of Sudia as disclosing adding combiner data in the form of a message control header that includes a timestamp. Fig. 18 shows that the message control header includes an optional timestamp, but does not disclose that this timestamp is related in any way to the time that an object was encrypted.

Claim 33 recites that the combiner data further includes a user ID associated with the user. The Examiner cited column 11, lines 33-39 and Fig. 18 of Sudia as disclosing adding combiner data in the form of a message control header that includes a an escrow certificate, which in turn includes a user ID in the form of a user name. According to Fig. 18, the message control header does not include a user escrow certificate. Rather, the header includes only the escrow certificate number. It would not be efficient to transmit the entire escrow certificate with an encrypted message, and Sudia does not disclose or suggest doing so.

The Examiner did not specifically address claims 34-37.

Claim 38 recites instructions to generate a second cryptographic key based at least in part on the at least one label split, and to encrypt the random split with the second cryptographic key, prior to adding the combiner data to the encrypted object. The random split included in the combiner data is the encrypted random split. The Examiner asserted numerous teachings of Sudia, none of which includes a disclosure that a second key is generated that is based on a label split used to generate the first key, or that the second key is used to encrypt the random split used to generate the first key, both prior to adding the combiner data to the encrypted object, and all as required by claim 38.

Claim 39 recites instructions to encrypt at least a portion of the combiner data with a header split before adding the combiner data to the encrypted object. Claim 40 recites that the header split is constant. The Examiner stated that Sudia teaches, at column 23, lines 27-31, that part of the header data is encrypted. In that passage, Sudia discloses that the sender's escrow certificate number is encrypted using the public

encryption key. Sudia does not disclose use of a header split for encrypting combiner data.

For at least the reasons noted above, the rejection of claims 27, 28, and 30-40 should be withdrawn.

The Examiner rejected claims 9 and 29 under 35 USC §103(a) as being unpatentable over Sudia, in view of Lipner et al., and further in view of Nguyen.

The Examiner relies on the teachings of Lipner et al. and Sudia as disclosing the elements of the claims as described above, and on Nguyen only for teaching that a key can be created from a user ID and password. The deficiencies of Lipner et al. and Sudia, and their combination, in disclosing the elements of independent claims 7 and 27 are discussed above. Nguyen does not overcome these deficiencies, nor does the Examiner assert that this is the case. Claims 9 and 29 depend from claims 7 and 27, respectively, and therefore no combination of the teachings of the asserted references could render obvious the invention as recited in claims 9 and 29. The rejection of claims 9 and 29, therefore, should be withdrawn.

New claims 41-60 are added. The subject matter of the new claims is disclosed in the written description, for example, in the passage spanning page 6, line 14 through page 11, line 18. New claims 41, 46, 51, and 56 recite that the combination of the plurality of key splits is a non-linear function. Thus, this claimed feature makes it clear that a split decrypt key does not necessarily imply that a corresponding encrypt key was formed from combined splits. Claims 42-45, 47-50, 52-55, and 57-60 recite other features of the claimed invention.

Based on the foregoing, it is submitted that all objections and rejections have been overcome. It is therefore requested that the Amendment be entered, the claims allowed, and the case passed to issue.

Respectfully submitted,

July 13, 2005
Date

Thomas M. Champagne
Registration No. 36,478
Customer Number 49691
(828) 253-8600

TMC:hlp